

# Activation et configuration de la fonctionnalité

## Introduction

La génération des vues de l'architecture est la responsabilité de l'addon "**nagvis-shinken-architecture**". Cet addon est une nouveauté de la version Shinken Entreprise V02.05.00.

Les sections suivantes décrivent le comportement de cet addon, comment l'activer/désactiver, et les différents paramètres de configuration disponibles.

## Sommaire

Introduction
Activation et désactivation de l'addon
Activation automatique
Activer/désactiver la fonctionnalité
Visualiser l'état de l'addon
Configuration de l'addon
Manipulation et étapes de configuration
Fonctionnement du module
Différents paramètres de configuration

## Activation et désactivation de l'addon

### Activation automatique

Lors d'une installation ou d'une mise à jour depuis une version antérieure à la V02.05.00, l'addon est automatiquement installé. Son activation dépend ensuite du type de machine sur laquelle Shinken est installé.

L'addon "**nagvis-shinken-architecture**" analyse la configuration reçue par l'Arbiter pour ensuite générer les vues graphiques des royaumes et les hôtes correspondant. Pour cette raison, il ne fait pas de sens que cette fonctionnalité soit activée sur une machine où aucun démon Arbiter n'est actif.

- Sur une machine n'ayant qu'un démon Poller actif par exemple, l'addon ne sera donc pas activé.
- Sur la machine principale d'une installation Shinken (c'est à dire celle qui contient l'Arbiter), l'addon "**nagvis-shinken-architecture**" sera automatiquement activé lors d'une installation ou mise à jour depuis une version antérieure à la V02.05.00.

### Activer/désactiver la fonctionnalité

Si cette fonctionnalité ne correspond pas à un besoin dans votre utilisation de Shinken Entreprise, elle peut bien sûr être désactivée et réactivée selon les envies.

Les sections suivantes décrivent de manière succincte les différentes commandes existantes permettant de manipuler les addons. Plus de détails sur ces commandes de manipulation des addons peuvent être trouvées dans la page de documentation dédiée:

### Vérifier l'état d'activation des addons

La commande "**shinken-addons-list**" permet de lister les addons ainsi que leur état d'activation. Dans l'exemple, on voit que l'addon "**nagvis-shinken-architecture**" est désactivé.

```
$ shinken-addons-list
nagvis                : ENABLED
nagvis-shinken-architecture : DISABLED
```

### Activer l'addon

Les addons peuvent être activés avec la commande "**shinken-addons-enable**". Par exemple, pour activer l'addon nagvis-shinken-architecture, la commande serait la suivante:

```
shinken-addons-enable nagvis-shinken-architecture
```

## Désactiver l'addon

Si on veut désactiver cet addon, une commande similaire permet de désactiver un addon. Dans le même exemple que précédemment, la commande est la suivante:

```
shinken-addons-disable nagvis-shinken-architecture
```

## Visualiser l'état de l'addon

L'état général de fonctionnement de l'addon peut être vérifié avec la commande **shinken-healthcheck**.

Dans le Healthcheck, l'état de l'addon est visible dans la section "Addons". Les points suivants sont vérifiés pour l'addon "nagvis-shinken-architecture":

- Présence du fichier de configuration correspondant (/etc/shinken/module/architecture-export.cfg)
- Présence de l'installation NagVis (logiciel responsable de l'affichage des architectures)
- L'interface d'affichage des informations de l'architecture est joignable.

```
Local addons [nagvis]
OK: NagVis installation found (/opt/nagvis)
'agvis' addon is running correctly at http://localhost/shinken-map
```

## Configuration de l'addon

## Manipulation et étapes de configuration

L'activation et désactivation de la fonctionnalité de visualisation de l'architecture Shinken s'effectue via l'addon "**nagvis-shinken-architecture**". La manipulation de l'addon est décrite dans la page [Accès à la fonctionnalité](#).

La configuration de cet addon s'effectue via le fichier de configuration du module "**architecture-export**", situé dans "*/etc/shinken/modules/architecture-export.cfg*".

## Fonctionnement du module

L'export de l'architecture s'effectue grâce au module "architecture-export". Le fonctionnement de ce module est séparé en 2 parties:

- **Envoi:** Lorsqu'un changement d'architecture est détecté, le module envoie la description de l'architecture de l'installation Shinken sur laquelle il est installé à une liste de modules destinataires.
- **Réception:** Le module reste également en attente de données. Lorsqu'il reçoit des données correspondant à une description d'architecture il génère les vues détaillées des royaumes et l'arbre des royaumes correspondants.

Dans le module, ces 2 aspects peuvent donc être configurés.

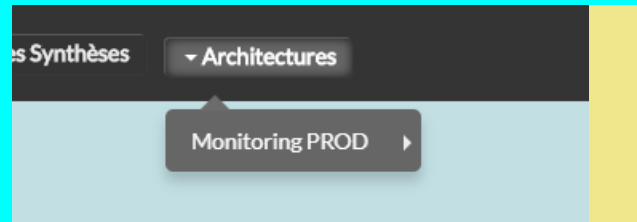
Aussi, le module est positionné sur l'Arbiter. On aura donc un seul module "architecture-export" par installation Shinken.

## Différents paramètres de configuration


## Nom de l'architecture

Lorsque le module "architecture-export" reçoit une description d'architecture, il faut pouvoir identifier l'architecture correspondante. Le paramètre **architecture\_name** permet donc de spécifier le nom de l'architecture qui sera visible dans l'interface de visualisation.

Ce nom est également présent dans les hôtes générés et donc également dans les cartes NagVis.



Lors d'une installation Shinken, ce nom est défini par défaut à :

 Shinken-<hostname\_machine\_arbiter>

### /etc/shinken/modules/architecture-export.cfg

```
define module {
    #==== Module identity =====
    # Module name. Must be unique
    module_name          architecture-export

    # Module type (to load module code). Do not edit.
    module_type          architecture_export

    .
    .
    .
    (contenu)
    .
    .
    .

    # Name with which this Shinken installation will be identified in the NagVis maps
    # The following characters are forbidden in the architecture name: ~!$%^&*"'|<>?,()=/+
    architecture_name    Monitoring PROD

    .
    .
    .
    (suite du fichier)
}
```

## Port et interface d'écoute du module

Lors d'un changement de l'architecture, la description de cette architecture est envoyée à un ou plusieurs autres modules. Il est possible de modifier les paramètres d'écoute du module, comme le port et l'interface d'écoute

#### /etc/shinken/modules/architecture-export.cfg

```
define module {
    .
    .
    .
    (contenu)
    .
    .
    .

    #==== Architecture description communication =====
    # This module opens a listening socket on which other shinken installations will send their architecture
    description.
    # When an architecture description is received by the module, it creates corresponding hosts and NagVis
    maps.

    # host: interface used for the listening socket (0.0.0.0 = all interfaces)
    host          0.0.0.0

    # Port to use for the listening socket
    port          7780

    .
    .
    .
    (suite du fichier)
}
```

## Sécurisation de la communication (SSL)

Il est également possible de forcer la connexion au module en HTTPS. Pour activer une communication sécurisée, il faut positionner le paramètre `use_ssl` à 1 (par défaut 0).

Pour utiliser son propre certificat, il faudra le spécifier avec les paramètres `ssl_cert` et `ssl_key`. Par défaut, le certificat livré avec Shinken est utilisé.

#### /etc/shinken/modules/architecture-export.cfg

```
define module {
    ...
    # 0 = Use HTTP, 1 = Use HTTPS
    use_ssl          1
    ssl_cert         /etc/shinken/certs/server.cert
    ssl_key          /etc/shinken/certs/server.key
    ...
}
```

## URL d'accès dans la visualisation

Dans l'interface de Visualisation, des liens sont présents dans la barre supérieure permettant un accès rapide à NagVis qui affiche l'architecture. Cette adresse est construite en fonction de l'adresse du démon Arbiter.

Dans certains cas, il se peut que cette adresse soit incorrecte. C'est le cas par exemple si l'adresse de l'Arbiter est l'adresse d'une interface locale utilisée pour la communication entre démons, tandis que NagVis et les interfaces de Configuration et Visualisation sont présentées sur une interface publique.

Dans le fichier de configuration du module, le paramètre `map_base_url` permet de spécifier la base de l'adresse utilisée pour la construction des adresses d'accès aux cartes. Pour être fonctionnelle, cette adresse doit correspondre à l'adresse de la machine qui héberge l'Arbiter.

#### /etc/shinken/modules/architecture-export.cfg

```
define module {
    ...
    # Base of URL used to display links in the Visualization UI
    # Defaults to Arbiter URL if empty
    # map_base_url http://example.com/
    ...
}
```



Ce paramètre peut être utilisé pour présenter un lien HTTPS vers NagVis (si NagVis a été configuré pour être servi en HTTPS dans Apache). Par exemple:

```
map_base_url https://example.com/
```

### Liste des destinataires

Enfin, lorsqu'un changement d'architecture est détecté, la description de l'architecture est envoyé à une liste de destinataires. Par défaut, le module **architecture-export** s'envoie les informations de l'architecture pour générer l'arbre des royaumes et les vues détaillées, mais il est également possible d'envoyer l'architecture à d'autres modules **"architecture-export"**. Il faudra, dans la liste des destinataires, prendre en compte le port ainsi que l'utilisation de SSL.

#### /etc/shinken/modules/architecture-export.cfg

```
define module {
    ...
    # Architecture description recipients
    # When the architecture of this Shinken installation changes (realms and daemons configuration),
    # and the arbiter is restarted the architecture description will be sent to the following hosts.
    send_my_architecture_to_recipients http://127.0.0.1:7780,https://adressemodule2:1234
    ...
}
```

Il n'est pas non plus obligatoire d'envoyer la description de l'architecture sur 127.0.0.1. Si on omet cette adresse dans la liste des destinataires, les vues détaillées et l'arbre des royaumes ne seront pas générées sur l'installation Shinken actuelle, mais seulement sur celles spécifiées dans la liste des destinataires.

Pour un cas d'application concret, voir la page [Exemple pratique: Mise en place automatisée d'une plateforme de supervision secondaire](#)